Homework 7

Christophe Hunt

March 18, 2017

Contents

1 Problem Set 1 **1**

1 Problem Set 1

This week, you'll have only one programming assignment. Please write a function to compute the expected value and standard deviation of an array of values. Compare your results with that of R's mean and std functions. Please document your work in an RMarkdown file and ensure that you have good comments to help the reader follow your work.

```
# vectors of different length

V1 <- c(5,9,3,23)
V2 <- c(10,11,12,13,14,15,10,13)
V3 <- c(1:20)

# vectors added to array
array_x <- array(c(V1,V2, V3))

# Expected results is the mean sum/count

expected_results <- function(x){
  return(sum(x)/dim(x)[1])
}

# Standard deviation of the sample population is the value less the mean, then squared,
#the sum of that divided by the length less 1 (adjustment made for sample population),
#then the square root of that final result

stand_dv_samp <- function(x){
  return(sqrt(sum((x - expected_results(x))^2)/(dim(x)[1] - 1)))
}
```

```
list("user defined mean function" = expected_results(array_x),
     "base R mean function" = mean(array_x))
```

```
## $`user defined mean function`
## [1] 10.875
##
## $`base R mean function`
## [1] 10.875
```

```
list("user defined sd function" = stand_dv_samp(array_x),
     "base R sd function" = sd(array_x))
```

```
## $`user defined sd function`
## [1] 5.545995
##
## $`base R sd function`
## [1] 5.545995
```

Now, consider that instead of being able to neatly fit the values in memory in an array, you have an infinite stream of numbers coming by. How would you estimate the mean and standard deviation of such a stream? Your function should be able to return the current estimate of the mean and standard deviation at any time it is asked. Your program should maintain these current estimates and return them back at any invocation of these functions. (Hint: You can maintain a rolling estimate of the mean and standard deviation and allow these to slowly change over time as you see more and more new values).

```
require(RSQLite)
```

```
## Loading required package: RSQLite
```

```
##set up a temporary sql database
x <- as.data.frame(array_x)
tmp <- dbConnect(SQLite(), ":memory:")
dbWriteTable(tmp, "temp_values", x)
```

```
## [1] TRUE
```

```
x <- c(2,1,4,5,6)
avg_sd_stream <- function(x){
                  suppressWarnings(dbWriteTable(tmp, "temp_values", as.data.frame(x), append = TRUE))
                  x <- dbGetQuery(tmp, "SELECT * FROM temp_values")
                  return(list("stream_avg" = expected_results(x), "stream_sd" = stand_dv_samp(x)))
}
```

```
#original results
list("original avg" = expected_results(array_x), "oringal sd" = stand_dv_samp(array_x))
```

```
## $`original avg`
## [1] 10.875
##
## $`oringal sd`
## [1] 5.545995
```

```
avg_sd_stream(x = c(1,2,35,12,41,15))
```

```
## $stream_avg
## [1] 11.94737
##
## $stream_sd
```

```
## [1] 8.372889
```

```
avg_sd_stream(x = c(2,7.6,100,2.67,129))
```

```
## $stream_avg
## [1] 16.16907
##
## $stream_sd
## [1] 23.64722
```

```
results <- list("samp_size" = dim(array_x)[1],
                "sum" = sum(array_x),
                "sum_sqrs" = sum(array_x^2),
                "last_mean" = mean(array_x),
                "last_sd" = sd(array_x))
results
```

```
## $samp_size
## [1] 32
##
## $sum
## [1] 348
##
## $sum_sqrs
## [1] 4738
##
## $last_mean
## [1] 10.875
##
## $last_sd
## [1] 5.545995
```

```
avg_sd_stream <- function(results, x){
                 results$samp_size <- results$samp_size + length(x)
                 results$sum <- results$sum + sum(x)
                 results$last_mean <- (results$sum/ results$samp_size)

                 results$sum_sqrs <- results$sum_sqrs + sum(x^2)

                 sigma_sq <- (results$sum_sqrs /
                                results$samp_size) - (results$last_mean^2)

                 results$last_sd <- sqrt(sigma_sq)
                 return(results)
}
results <- avg_sd_stream(results, c(1,2,4,5,6,7))
results
```

```
## $samp_size
## [1] 38
##
## $sum
## [1] 373
##
## $sum_sqrs
## [1] 4869
```

```
##
## $last_mean
## [1] 9.815789
##
## $last_sd
## [1] 5.63754
```

```
results <- avg_sd_stream(results, c(1,2,4,5,6,7))
results
```

```
## $samp_size
## [1] 44
##
## $sum
## [1] 398
##
## $sum_sqrs
## [1] 5000
##
## $last_mean
## [1] 9.045455
##
## $last_sd
## [1] 5.640578
```